|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  **«Российский технологический университет - МИРЭА»** |
| **РТУ МИРЭА** |

Институт информационных технологий (ИИТ)Кафедра инструментального и прикладного программного обеспечения (ИиППО)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №3**

**по дисциплине**

«Инструментальное программное обеспечение разработки и проектирования информационных систем»

Выполнил студент группы ИКМО-01-20 Реина Эрнандес Ранди

Фамилия И.О.

Принял(а) Куликов Александр Анатольевич Фамилия И.О.

Работы выполнены «\_\_»\_\_\_\_\_\_\_2021 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

«Зачтено» «\_\_»\_\_\_\_\_\_\_2021 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Москва, 2021г.

**Оглавление**

[**Задание** 3](#_Toc65605745)

[**Код программы с описанием работы** 3](#_Toc65605746)

[**Скриншоты работы** 13](#_Toc65605747)

[**Заключение** 13](#_Toc65605748)

# **Задание**

Создать приложение по отправки email писем из приложения разработанного на Python.

Задачи:

Доработать приложение по отправки email писем

Необходимо сделать массовую отправку адресатам со вложениями.

# **Код программы с описанием работы**

Для начала разработаем форму с помощью QT Designer. (Рис.1.)

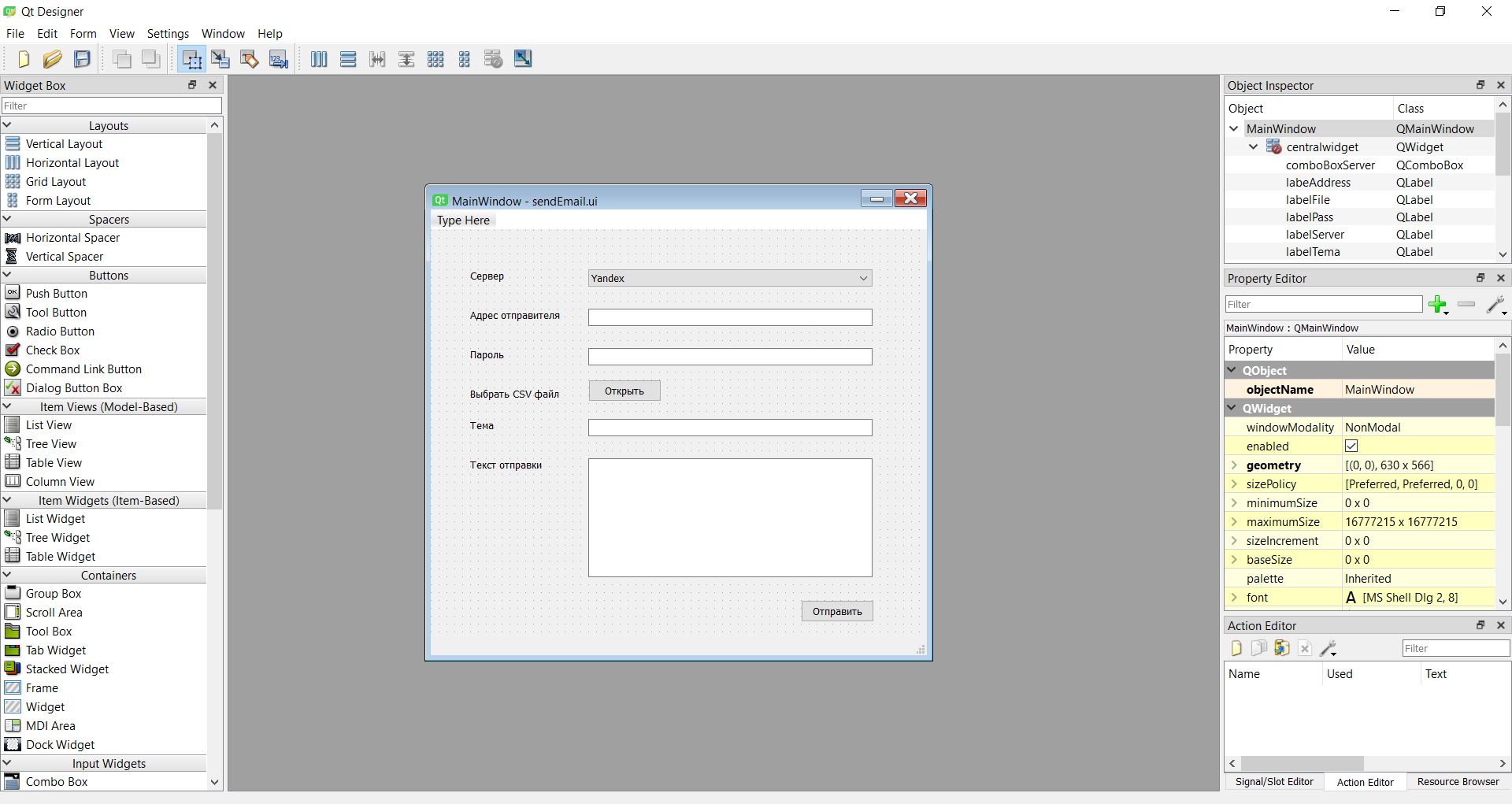


Рис.1 – Форма отправки email писем

Чтобы иметь возможность массово отправлять электронные письма с вложениями, был сделан CSV-файл. В CSV-файле первое значение — это "адресат", а следующее -"; " имена файлов (адрес файла). Пример содержимого этого файла:

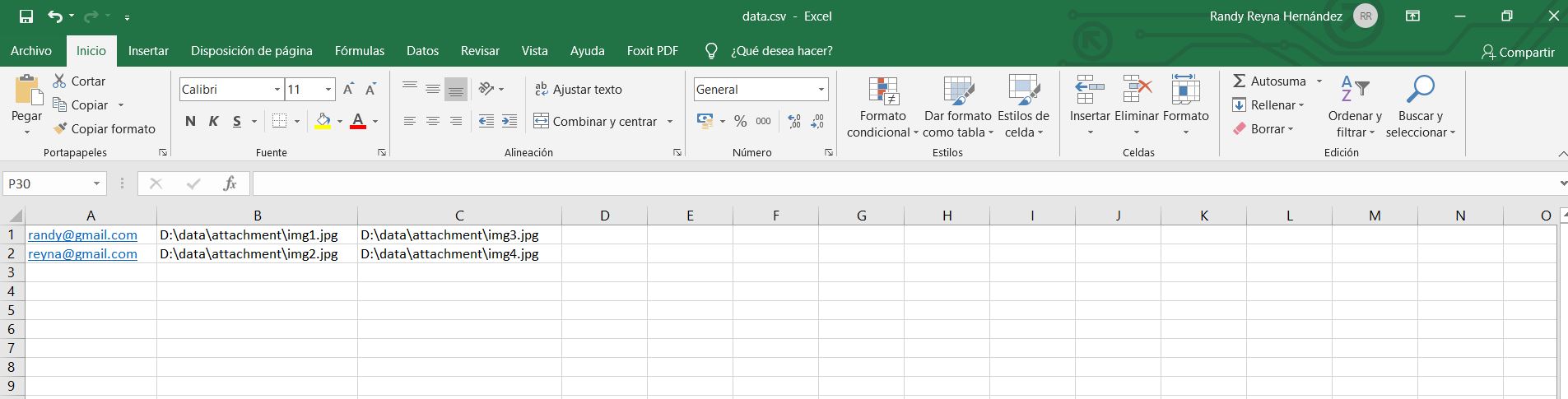


Рис.2 – файл CSV

**send\_email.ui**

Затем мы сохраняем форму QT Designer в файл .ui

<?xml version="1.0" encoding="UTF-8"?>

<ui version="4.0">

 <class>MainWindow</class>

 <widget class="QMainWindow" name="MainWindow">

  <property name="geometry">

   <rect>

    <x>0</x>

    <y>0</y>

    <width>630</width>

    <height>566</height>

   </rect>

  </property>

  <property name="windowTitle">

   <string>MainWindow</string>

  </property>

  <widget class="QWidget" name="centralwidget">

   <widget class="QLabel" name="labelServer">

    <property name="geometry">

     <rect>

      <x>50</x>

      <y>50</y>

      <width>55</width>

      <height>16</height>

     </rect>

    </property>

    <property name="text">

     <string>Сервер</string>

    </property>

   </widget>

   <widget class="QLabel" name="labeAddress">

    <property name="geometry">

     <rect>

      <x>50</x>

      <y>100</y>

      <width>121</width>

      <height>16</height>

     </rect>

    </property>

    <property name="text">

     <string>Адрес отправителя</string>

    </property>

   </widget>

   <widget class="QLabel" name="labelPass">

    <property name="geometry">

     <rect>

      <x>50</x>

      <y>150</y>

      <width>55</width>

      <height>16</height>

     </rect>

    </property>

    <property name="text">

     <string>Пароль</string>

    </property>

   </widget>

   <widget class="QLabel" name="labelFile">

    <property name="geometry">

     <rect>

      <x>50</x>

      <y>200</y>

      <width>121</width>

      <height>16</height>

     </rect>

    </property>

    <property name="text">

     <string>Выбрать CSV файл</string>

    </property>

   </widget>

   <widget class="QLabel" name="labelTema">

    <property name="geometry">

     <rect>

      <x>50</x>

      <y>240</y>

      <width>55</width>

      <height>16</height>

     </rect>

    </property>

    <property name="text">

     <string>Тема</string>

    </property>

   </widget>

   <widget class="QLabel" name="labelText">

    <property name="geometry">

     <rect>

      <x>50</x>

      <y>290</y>

      <width>111</width>

      <height>16</height>

     </rect>

    </property>

    <property name="text">

     <string>Текст отправки</string>

    </property>

   </widget>

   <widget class="QLineEdit" name="lineEditAddress">

    <property name="geometry">

     <rect>

      <x>200</x>

      <y>100</y>

      <width>361</width>

      <height>22</height>

     </rect>

    </property>

   </widget>

   <widget class="QLineEdit" name="lineEditPass">

    <property name="geometry">

     <rect>

      <x>200</x>

      <y>150</y>

      <width>361</width>

      <height>22</height>

     </rect>

    </property>

   </widget>

   <widget class="QPushButton" name="pushButtonOpenFile">

    <property name="geometry">

     <rect>

      <x>200</x>

      <y>190</y>

      <width>93</width>

      <height>28</height>

     </rect>

    </property>

    <property name="text">

     <string>Открыть</string>

    </property>

   </widget>

   <widget class="QComboBox" name="comboBoxServer">

    <property name="geometry">

     <rect>

      <x>200</x>

      <y>50</y>

      <width>361</width>

      <height>22</height>

     </rect>

    </property>

    <item>

     <property name="text">

      <string>Yandex</string>

     </property>

    </item>

    <item>

     <property name="text">

      <string>Mail</string>

     </property>

    </item>

    <item>

     <property name="text">

      <string>Gmail</string>

     </property>

    </item>

   </widget>

   <widget class="QLineEdit" name="lineEditTema">

    <property name="geometry">

     <rect>

      <x>200</x>

      <y>240</y>

      <width>361</width>

      <height>22</height>

     </rect>

    </property>

   </widget>

   <widget class="QTextEdit" name="textEditText">

    <property name="geometry">

     <rect>

      <x>200</x>

      <y>290</y>

      <width>361</width>

      <height>151</height>

     </rect>

    </property>

   </widget>

   <widget class="QPushButton" name="pushButtonSend">

    <property name="geometry">

     <rect>

      <x>470</x>

      <y>470</y>

      <width>93</width>

      <height>28</height>

     </rect>

    </property>

    <property name="text">

     <string>Отправить</string>

    </property>

   </widget>

  </widget>

  <widget class="QMenuBar" name="menubar">

   <property name="geometry">

    <rect>

     <x>0</x>

     <y>0</y>

     <width>630</width>

     <height>26</height>

    </rect>

   </property>

  </widget>

  <widget class="QStatusBar" name="statusbar"/>

 </widget>

 <resources/>

 <connections/>

</ui>

**send\_email.ui.py**

После этого мы преобразуем файл .ui в .py

# -\*- coding: utf-8 -\*-

# Form implementation generated from reading ui file 'sendEmail.ui'

#

# Created by: PyQt5 UI code generator 5.15.2

#

# WARNING: Any manual changes made to this file will be lost when pyuic5 is

# run again.  Do not edit this file unless you know what you are doing.

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui\_MainWindow(object):

    def setupUi(self, MainWindow):

        MainWindow.setObjectName("MainWindow")

        MainWindow.resize(630, 566)

        self.centralwidget = QtWidgets.QWidget(MainWindow)

        self.centralwidget.setObjectName("centralwidget")

        self.labelServer = QtWidgets.QLabel(self.centralwidget)

        self.labelServer.setGeometry(QtCore.QRect(50, 50, 55, 16))

        self.labelServer.setObjectName("labelServer")

        self.labeAddress = QtWidgets.QLabel(self.centralwidget)

        self.labeAddress.setGeometry(QtCore.QRect(50, 100, 121, 16))

        self.labeAddress.setObjectName("labeAddress")

        self.labelPass = QtWidgets.QLabel(self.centralwidget)

        self.labelPass.setGeometry(QtCore.QRect(50, 150, 55, 16))

        self.labelPass.setObjectName("labelPass")

        self.labelFile = QtWidgets.QLabel(self.centralwidget)

        self.labelFile.setGeometry(QtCore.QRect(50, 200, 121, 16))

        self.labelFile.setObjectName("labelFile")

        self.labelTema = QtWidgets.QLabel(self.centralwidget)

        self.labelTema.setGeometry(QtCore.QRect(50, 240, 55, 16))

        self.labelTema.setObjectName("labelTema")

        self.labelText = QtWidgets.QLabel(self.centralwidget)

        self.labelText.setGeometry(QtCore.QRect(50, 290, 111, 16))

        self.labelText.setObjectName("labelText")

        self.lineEditAddress = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEditAddress.setGeometry(QtCore.QRect(200, 100, 361, 22))

        self.lineEditAddress.setObjectName("lineEditAddress")

        self.lineEditPass = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEditPass.setGeometry(QtCore.QRect(200, 150, 361, 22))

        self.lineEditPass.setObjectName("lineEditPass")

        self.pushButtonOpenFile = QtWidgets.QPushButton(self.centralwidget)

        self.pushButtonOpenFile.setGeometry(QtCore.QRect(200, 190, 93, 28))

        self.pushButtonOpenFile.setObjectName("pushButtonOpenFile")

        self.comboBoxServer = QtWidgets.QComboBox(self.centralwidget)

        self.comboBoxServer.setGeometry(QtCore.QRect(200, 50, 361, 22))

        self.comboBoxServer.setObjectName("comboBoxServer")

        self.comboBoxServer.addItem("")

        self.comboBoxServer.addItem("")

        self.comboBoxServer.addItem("")

        self.lineEditTema = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEditTema.setGeometry(QtCore.QRect(200, 240, 361, 22))

        self.lineEditTema.setObjectName("lineEditTema")

        self.textEditText = QtWidgets.QTextEdit(self.centralwidget)

        self.textEditText.setGeometry(QtCore.QRect(200, 290, 361, 151))

        self.textEditText.setObjectName("textEditText")

        self.pushButtonSend = QtWidgets.QPushButton(self.centralwidget)

        self.pushButtonSend.setGeometry(QtCore.QRect(470, 470, 93, 28))

        self.pushButtonSend.setObjectName("pushButtonSend")

        MainWindow.setCentralWidget(self.centralwidget)

        self.menubar = QtWidgets.QMenuBar(MainWindow)

        self.menubar.setGeometry(QtCore.QRect(0, 0, 630, 26))

        self.menubar.setObjectName("menubar")

        MainWindow.setMenuBar(self.menubar)

        self.statusbar = QtWidgets.QStatusBar(MainWindow)

        self.statusbar.setObjectName("statusbar")

        MainWindow.setStatusBar(self.statusbar)

        self.retranslateUi(MainWindow)

        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):

        \_translate = QtCore.QCoreApplication.translate

        MainWindow.setWindowTitle(\_translate("MainWindow", "MainWindow"))

        self.labelServer.setText(\_translate("MainWindow", "Сервер"))

        self.labeAddress.setText(\_translate("MainWindow", "Адрес отправителя"))

        self.labelPass.setText(\_translate("MainWindow", "Пароль"))

        self.labelFile.setText(\_translate("MainWindow", "Выбрать CSV файл"))

        self.labelTema.setText(\_translate("MainWindow", "Тема"))

        self.labelText.setText(\_translate("MainWindow", "Текст отправки"))

        self.pushButtonOpenFile.setText(\_translate("MainWindow", "Открыть"))

        self.comboBoxServer.setItemText(0, \_translate("MainWindow", "Yandex"))

        self.comboBoxServer.setItemText(1, \_translate("MainWindow", "Mail"))

        self.comboBoxServer.setItemText(2, \_translate("MainWindow", "Gmail"))

        self.pushButtonSend.setText(\_translate("MainWindow", "Отправить"))

if \_\_name\_\_ == "\_\_main\_\_":

    import sys

    app = QtWidgets.QApplication(sys.argv)

    MainWindow = QtWidgets.QMainWindow()

    ui = Ui\_MainWindow()

    ui.setupUi(MainWindow)

    MainWindow.show()

    sys.exit(app.exec\_())

**app.py**

Для того, чтобы была возможность отправлять email письма, был сделан обработчик события «click» по кнопке отправить. Для этого создан коннект и функция send\_email:

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

#

# https://pythones.net

import sys

import smtplib

import mimetypes

import os

import csv

try:

    from email import encoders

except:

    from email import Encoders as encoders

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.audio import MIMEAudio

from email.mime.image import MIMEImage

from email.mime.base import MIMEBase

from PyQt5 import uic, QtWidgets

from PyQt5.QtWidgets import QFileDialog

from PyQt5 import QtCore, QtGui, QtWidgets

qtCreatorFile = "sendEmail.ui"

Ui\_MainWindow, QtBaseClass = uic.loadUiType(

    qtCreatorFile)

class VentanaPrincipal(QtWidgets.QMainWindow, Ui\_MainWindow):

    msg\_str = []

    def \_\_init\_\_(self):

        QtWidgets.QMainWindow.\_\_init\_\_(self)

        Ui\_MainWindow.\_\_init\_\_(self)

        self.setupUi(self)

        self.pushButtonSend.clicked.connect(self.send\_email)

        self.pushButtonOpenFile.clicked.connect(self.open\_file)

    def send\_email(self):

        if(self.msg\_str):

            address = self.lineEditAddress.text()

            password = self.lineEditPass.text()

            if self.comboBoxServer.currentIndex() == 0:

                server = smtplib.SMTP('smtp.yandex.ru:465')

            elif self.comboBoxServer.currentIndex() == 1:

                server = smtplib.SMTP('smtp.mail.ru:25')

            elif self.comboBoxServer.currentIndex() == 2:

                server = smtplib.SMTP('smtp.gmail.com:587')

            server.starttls()

            server.login(address, password)

            for msg in self.msg\_str:

                from\_addr = msg['From']

                to\_addr = msg['To']

                server.sendmail(from\_addr, to\_addr, msg.as\_string())

            server.quit()

        else:

            print("Сначала выберите файл")

    def prepare\_email(self, address, files):

        tema = self.lineEditTema.text()

        text = self.textEditText.toPlainText()

        msg = MIMEMultipart()

        msg['From'] = self.lineEditAddress.text()

        msg['To'] = address

        msg['Subject'] = tema

        msg.attach(MIMEText(text, 'plain'))

        self.process\_attachement(msg, files)

        self.msg\_str.append(msg)

    def open\_file(self):

        options = QFileDialog.Options()

        options |= QFileDialog.DontUseNativeDialog

        fileName, \_ = QFileDialog.getOpenFileName(

            self, "QFileDialog.getOpenFileName()", "", "csv (\*.csv);;All Files (\*)", options=options)

        if fileName:

            with open(fileName, "r", encoding="utf-8-sig") as File:

                reader = csv.reader(File)

                for row in reader:

                    files = []

                    str\_row = row[0].split(";")

                    files.append(str\_row[1])

                    files.append(str\_row[2])

                    self.prepare\_email(str(str\_row[0]), files)

    def process\_attachement(self, msg, files):

        for f in files:

            if os.path.isfile(f):

                self.attach\_file(msg, f)

            elif os.path.exists(f):

                dir = os.listdir(f)

                for file in dir:

                    self.attach\_file(msg, f+"/"+file)

    def attach\_file(self, msg, filepath):

        filename = os.path.basename(filepath)

        ctype, encoding = mimetypes.guess\_type(filepath)

        if ctype is None or encoding is not None:

            ctype = 'application/octet-stream'

        maintype, subtype = ctype.split('/', 1)

        if maintype == 'text':

            with open(filepath) as fp:

                file = MIMEText(fp.read(), \_subtype=subtype)

                fp.close()

        elif maintype == 'image':

            with open(filepath, 'rb') as fp:

                file = MIMEImage(fp.read(), \_subtype=subtype)

                fp.close()

        elif maintype == 'audio':

            with open(filepath, 'rb') as fp:

                file = MIMEAudio(fp.read(), \_subtype=subtype)

                fp.close()

        else:

            with open(filepath, 'rb') as fp:

                file = MIMEBase(maintype, subtype)

                file.set\_payload(fp.read())

                fp.close()

                encoders.encode\_base64(file)

        file.add\_header('Content-Disposition', 'attachment',

                        filename=filename)

        msg.attach(file)

if \_\_name\_\_ == "\_\_main\_\_":

    app = QtWidgets.QApplication([])

    ventana = VentanaPrincipal()

    ventana.show()

    app.exec\_()

Создана коннект и событие по нажатию на Button, для открытия диалогового окна с возможностью выбора файла CSV (функция «open\_file»).

Функция «open\_file» готовит электронную почту со всеми получателями и вложениями через функцию «prepare\_email»

Для добавления файлов в письмо, была написана функция по обработке списка, добавляемых к сообщению файлов и функция по добавлению конкретного файла к сообщению (функции «process\_attachement» и «attach\_file»).

# **Скриншоты работы**

На рисунке 3 показан снимок экрана запущенного приложения.

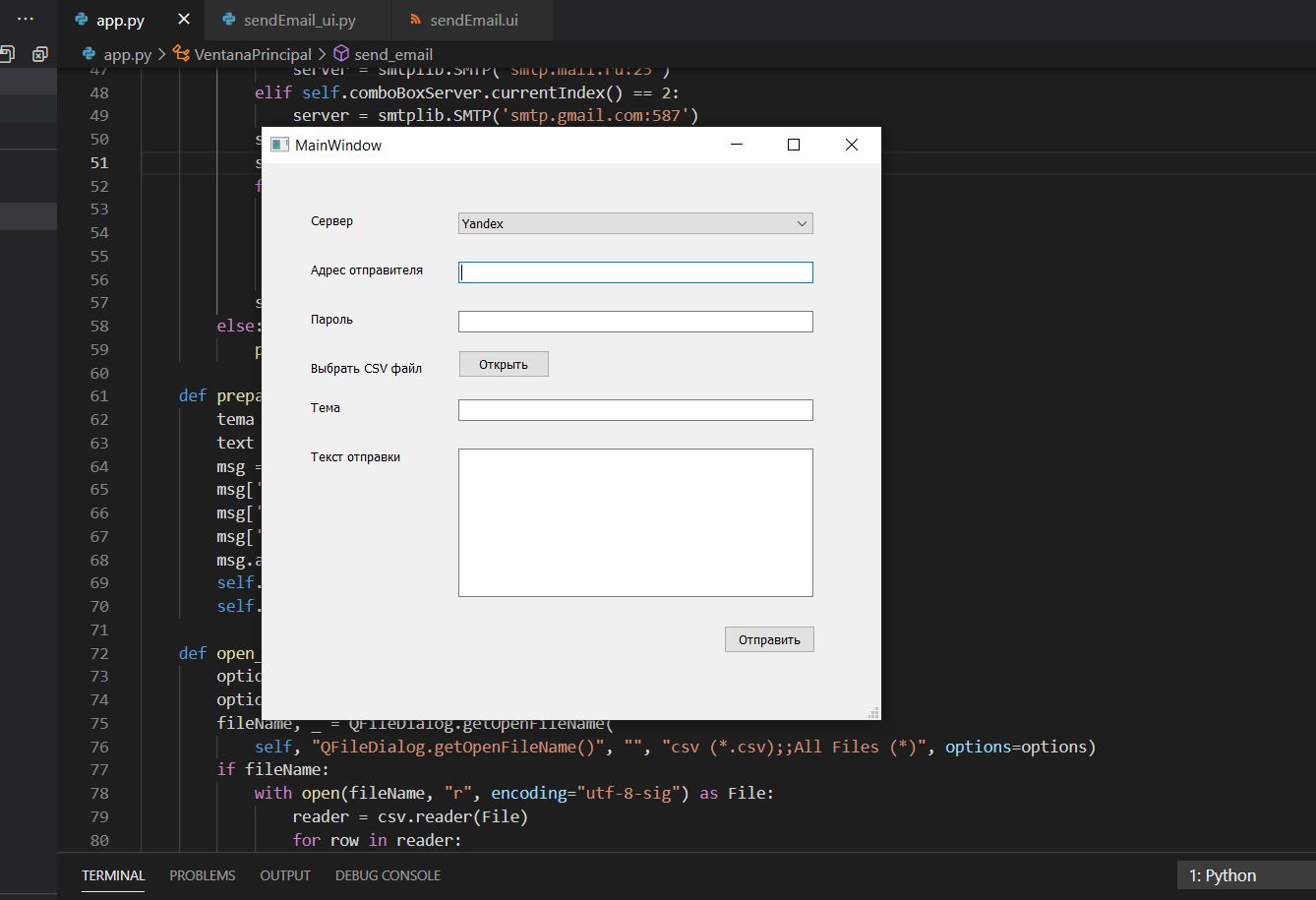


Рис.3 – Запущенное приложение.

# **Заключение**

В ходе практической работы были закреплены знания языка Python, а также была освоена технология массовой отправки писем с вложениями. Чтение CSV файла. Было написано полностью рабочее приложение по отправки писем.